VII International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2017
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

A DIRECT SOLUTION APPROACH FOR MULTI TIMESCALE OPTIMAL CONTROL PROBLEMS

MATTHIAS BITTNER*, BENEDIKT GRÜTER*, JOHANNES DIEPOLDER* AND FLORIAN HOLZAPFEL*

* Institute of Flight System Dynamics (FSD)
Technical University of Munich (TUM)
Boltzmannstraße 15, 85748 Garching, Germany
e-mail: m.bittner@tum.de, benedikt.grueter@tum.de,
johannes.diepolder@tum.de, florian.holzapfel@tum.de
web page: http://www.fsd.mw.tum.de/

Key words: Optimal Control, Direct Discretization, Multiple Timescales, Stiff Dynamics, Aircraft Trajectory Optimization

Abstract. In high fidelity optimal control problems, a commonly appearing problem emerges from different timescales inherent to the model, resulting in stiff differential equations. When solving these problems using direct discretization, the selection of the discretization nodes for all states is driven by the states associated with the fast dynamics, no matter how strong their influence on the solution is. In this paper, a novel discretization scheme is presented that uses direct collocation for the slow states while the fast states of the model are represented based on a direct multiple shooting scheme. This way, different grids may be chosen for the states, resulting in a slight decoupling of the timescales. A high fidelity air race trajectory optimization problem is implemented to demonstrate how the dimensions of the discretized problem can be significantly decreased by the method, resulting in improved computational performance during the solution process.

1 INTRODUCTION

In many applications, the performance of a system needs to be increased without changing its inherent properties. In these cases, methods for the optimization of operation strategies are required, where optimal control is one such method. When applying optimal control theory to high fidelity models, a problem that appears quite often are different timescales inherent to the model, resulting in *stiff differential equations*.

Especially in mechanical systems, fast dynamics can often be recognized as (small scale) *internal dynamics*, while slower dynamics often represent the more visible (large scale) *outer effects* of the motion of a body. If the model may not be simplified by assuming the fast part of the dynamics to be decayed instantaneously, the selection of an appropriate

discretization grid is driven by the fast and small scale dynamics (in order to keep the integration errors low) whose dynamic effects hardly influence the overall results. In many high fidelity optimal control problems, the required fine grid strongly increases the computational effort of the solution process.

Several approaches have been suggested to overcome this issue in the past. These include *Multirate Runge Kutta* methods [1, 2] that may also be used for pure simulation tasks. Furthermore, a multi timescale collocation method has been presented in [3]. BOTTASSO et. al. have published results based on the combination of single and multiple shooting in [4]. In their work, a discretization scheme is proposed that combines the benefits of *Direct Single Shooting* with those of *Direct Multiple Shooting*.

The method proposed here, is to split the state vector, and thus the dynamic system, into a fast and a slow part. The fast dynamics are discretized using *Multiple Shooting* and the slow ones using *Direct Collocation* that is based on the same grid as the multiple shooting *segment nodes*. The number of Multiple Shooting integration grid points in between these segment nodes may be chosen arbitrarily – forming a second, finer grid to coexist with the collocation grid. In order to be able to evaluate the dynamic equations within the multiple shooting segments, an approximation of the slow states is required that can be calculated based on interpolation.

The presented approach is applied to an aircraft trajectory optimization problem, using a nonlinear high fidelity rigid body simulation model of an aerobatic aircraft. The model features very fast rotational dynamics compared to the translational motion. In this example, a time optimal trajectory through an air race course respecting several boundary conditions and path constraints is calculated. The results show a significant reduction in the dimensions of the numerical optimization problem, also leading to a reduced calculation time.

The method and the results published in this paper are part of the dissertation thesis [5] of the first author.

2 OPTIMAL CONTROL PROBLEM

The problems considered here are optimal control problems of the following form: Determine the optimal control histories $\mathbf{u}_{opt}(t) \in \mathbb{R}^{n_u}$ and the optimal state trajectory $\mathbf{x}_{opt}(t) \in \mathbb{R}^{n_x}$ that minimize the BOLZA cost functional

$$J = e(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \tag{1}$$

subject to the state dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{2}$$

the initial and final boundary conditions

$$\boldsymbol{\psi}_0(\mathbf{x}(t_0), t_0) = \mathbf{0} \qquad \text{and} \qquad \boldsymbol{\psi}_f(\mathbf{x}(t_f), t_f) = \mathbf{0}, \tag{3}$$

and the equality and inequality path constraints

$$\mathbf{C}_{eq}(\mathbf{x}(t), \mathbf{u}(t), t) = \mathbf{0} \qquad \text{and} \qquad \mathbf{C}_{ineq}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0}. \tag{4}$$

The problem in the example below is formulated as a multi phase problem. [5, 6]

2.1 Discretization Techniques

Two of the classes of methods for solving optimal control problems are: On the one hand, there are indirect approaches that are based on the derivation of optimality conditions for the continuous problem which are then discretized and solved in a second step. On the other hand, in direct schemes the problem is first discretized and afterwards optimized numerically. The process of discretization turns the infinite dimensional optimal control problem into a finite dimensional numerical optimization problem that can be solved using off-the-shelf optimization algorithms like gradient based *Sequential Quadratic Programming (SQP)* or *Interior Point (IP)* methods.

The two direct solution approaches combined here are *Direct Multiple Shooting* and *Direct Collocation*. In shooting, the dynamic constraint (2) is discretized by performing a forward integration using numerical simulation methods for ordinary differential equations, like Runge Kutta methods. If the simulation of the whole time series is performed in one sweep, the methods are called *Single Shooting* methods. Opposite, in *Multiple Shooting* methods the simulation is reset at some predefined nodes, the so-called *Multiple Shooting Defect Nodes*. The state values at these nodes are introduced as optimization variables in the discretized problem. In order to ensure continuous and feasible state trajectories, for each multiple shooting node an additional set of constraints, the so-called *Multiple Shooting Defects*, are introduced:

$$\mathbf{c}_m = \mathbf{x}_0^m(\mathbf{z}) - \mathbf{x}_0^{m+1}(\mathbf{z}) + \int_{t_{\mathbf{x}_0,m}}^{t_{\mathbf{x}_0,m+1}} \dot{\mathbf{x}}(t)\, dt \qquad \forall m = 1, \ldots, M-1 \tag{5}$$

The gradient information required by the numerical optimization algorithms can be calculated by finite differences or automatic differentiation and sensitivity equations. More details on the latter can be found in literature such as [5, 6].

One way of motivating *Direct Collocation* is by resetting the state values in a multiple shooting method after each integration step. Then, every discretized state value is part of the optimization parameter vector in the numerical optimization problem. Moreover, a collocation defect needs to be introduced for every discretization time step:

$$\mathbf{c}_k = \mathbf{x}_k(\mathbf{z}) - \mathbf{x}_{k+1}(\mathbf{z}) + h_k \cdot \Phi(\mathbf{x}_k(\mathbf{z}), \mathbf{x}_{k+1}(\mathbf{z}), \mathbf{u}_k(\mathbf{z}), \mathbf{u}_{k+1}(\mathbf{z}), \mathbf{p}(\mathbf{z})) \tag{6}$$

Here, Φ represents the increment function of the underlying integration scheme. Remarkably, in collocation methods the use of implicit integration schemes is comparably easy

as all state values are part of the optimization vector and available for the calculation of the defects. Consequently, the solution of the implicit integration scheme is performed by the numerical optimization algorithm automatically. More details on direct and indirect solution methods for optimal control problems can also be found in [6, 5].

3 DYNAMIC AIRCRAFT MODEL

A full nonlinear rigid body simulation model of an aerobatic aircraft incorporating nonlinear aerodynamics is used here. Table 1 lists the states of the model, which is controlled by the deflections of the aileron ξ, the elevator η, the rudder ζ and the thrust lever position $\delta_{T,CMD}$. No wind is considered in the example, rendering all aerodynamic and kinematic quantities equal.

The considered flight trajectory is of small spatial extent, the flight times are comparably short, and the velocities are low enough, such that the earth may be considered as flat and non-rotating. In this case, the position equations of motion for the aircraft can be given with respect to a locally fixed coordinate frame. The attitude dynamics are modeled in accordance to the work [7], given with respect to the kinematic flight path of the aircraft. The rotational and the translational dynamics of the rigid body are based on NEWTON's second law using the moments and forces acting on the system as inputs.

The main categories of external forces and moments influencing the motion of an aircraft are aerodynamics, propulsion and gravity. Here, the thrust dynamics are approximated by a first order system in δ_T. More information on modeling the external forces and moments can e. g. be found in [7, 8, 9, 10, 11].

As the flights are partially conducted in the proximity of stall, a model for the lift coefficient that features a linear dependency on the angle of attack is not sufficient. Hence, a nonlinear function $C^*_{L,\alpha}(\alpha_K)$ is used for modeling its dependence on the angle of attack. Moreover, the propulsion force is assumed to act along the longitudinal axis of the aircraft. As the thrust created by the engine cannot follow the commanded thrust instantaneously, a thrust dynamic is incorporated in the model. Due to the small spatial extent of the trajectories a gravitational model assuming constant acceleration is used. [5]

Table 1: States of the aircraft simulation model

POSITION			ATTITUDE		
Symbol	**Description**	**Unit**	**Symbol**	**Description**	**Unit**
$(x)_N$	x-position in N Frame	m	α_K	Angle of attack	rad
$(y)_N$	y-position in N Frame	m	β_K	Angle of sideslip	rad
$(z)_N$	Downward position	m	μ_K	Bank angle	rad
VELOCITY			ANGULAR RATE		
Symbol	**Description**	**Unit**	**Symbol**	**Description**	**Unit**
V_K	Absolute velocity	m/s	p	Roll rate	rad/s
χ_K	Course angle	rad	q	Pitch rate	rad/s
γ_K	Climb angle	rad	r	Yaw rate	rad/s

4 COMBINED DIRECT COLLOCATION AND MULTIPLE SHOOTING

The proposed approach for overcoming the aforementioned issues combines a direct collocation scheme with a direct multiple shooting approach. The basic idea is to separate the state vector of the dynamic system into two parts, one containing the fast and the other containing the slow dynamics. The fast dynamics are discretized using a multiple shooting approach and the slow ones using a collocation scheme that is based on the same grid as the *Multiple Shooting Defect Nodes*. The number of multiple shooting integration grid points in between these nodes may be chosen arbitrarily, forming another, finer grid. In order to be able to evaluate the dynamic equations in between the multiple shooting segments, an approximation of the slow states is required. These values can be calculated using different interpolation algorithms, where a cubic and a linear interpolation scheme is suggested here. The required gradients can be evaluated using slightly modified sensitivity equations.

Figure 1 visualizes the general idea of the discretization scheme. The two grids inherent to the method can be seen on the abscissa with the large markers representing the collocation and *Multiple Shooting Defect Node* grid

$$\mathbb{G}_{\mathbf{x}_s} = \{t_0 < t_1 < t_2 < \ldots < t_N = t_f\} \tag{7}$$

and the small markers representing the *Multiple Shooting Integration Grid*:

$$\mathbb{G}_{\mathbf{x}_f} = \{\bar{\tau}_{0,0}, \bar{\tau}_{0,1}, \ldots, \bar{\tau}_{0,n_0} = \bar{\tau}_{1,0}, \bar{\tau}_{1,1}, \ldots, \bar{\tau}_{1,n_1} = \bar{\tau}_{2,0}, \ldots, \bar{\tau}_{N,n_N}\} \tag{8}$$

In the following, they will be referred to as the *coarse grid* $\mathbb{G}_{\mathbf{x}_s}$ and the *fine grid* $\mathbb{G}_{\mathbf{x}_f}$. The multiple shooting defect between the k-th and the $k+1$-th multiple shooting segment is visualized as $d_{f,k}$.

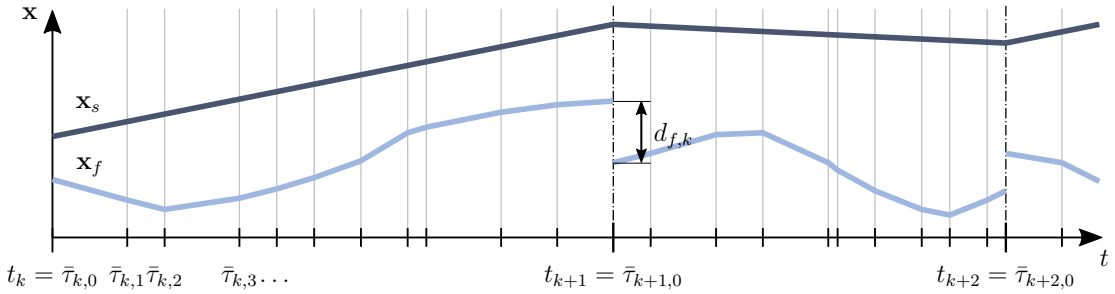

Figure 1: Discretization scheme based on collocation and multiple shooting. [5]

The dynamics of the system are partitioned into a slow part and a fast part by separating the state vector $\mathbf{x}$ into the sub-vector $\mathbf{x}_s$ for the slow states (shown in dark blue in figure 1) and $\mathbf{x}_f$ for the fast states (depicted in light blue in figure 1). Consequently, the state dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \tag{9}$$

also need to be partitioned into the fast dynamics $\mathbf{f}_f$ and the slow dynamics $\mathbf{f}_s$. Equation (9) can hence be rewritten as:

$$\dot{\mathbf{x}}_s = \mathbf{f}_s(\mathbf{x}_s, \mathbf{x}_f, \mathbf{u}, \mathbf{p}), \qquad \text{and} \qquad \dot{\mathbf{x}}_f = \mathbf{f}_f(\mathbf{x}_s, \mathbf{x}_f, \mathbf{u}, \mathbf{p}) \tag{10}$$

Now, the evaluation of the slow dynamics on the coarse grid is directly possible as all required data is part of the numerical optimization vector in the discretized problem. On the contrary, the slow states are not available on the fine grid. At this point, the aforementioned linear or cubic interpolation schemes are used. For both of them, the normalized time $\bar{\tau}_k \in [0, 1]$ is introduced in each segment $[t_k, t_{k+1}]$ of the coarse grid.

The linear interpolation of the fast states can directly be calculated from the initial and the final state values of each multiple shooting segment (with $\mathbf{x}_s(t_k) = \mathbf{x}_{s,k}$):

$$\mathbf{x}_s(\bar{\tau}_k) = \mathbf{x}_{s,k} + \bar{\tau}_k \cdot (\mathbf{x}_{s,k+1} - \mathbf{x}_{s,k}) \tag{11}$$

When using a cubic interpolation in the segments, the state derivatives at the initial and the final point also need to be considered. They can be calculated from the following state equations:

$$\dot{\mathbf{x}}_{s,k} = \mathbf{f}_s(\mathbf{x}_{s,k}, \mathbf{x}_{f,k}, \mathbf{u}_k, \mathbf{p}) = \mathbf{f}_{s,k} \tag{12}$$

$$\dot{\mathbf{x}}_{s,k+1} = \mathbf{f}_s(\mathbf{x}_{s,k+1}, \mathbf{x}_{f,k+1}, \mathbf{u}_{k+1}, \mathbf{p}) = \mathbf{f}_{s,k+1} \tag{13}$$

Requiring the state values and the state derivatives to match at the initial and the final point of each segment, the following cubic polynomial can be derived:

$$\begin{aligned} \mathbf{x}_s(\bar{\tau}_k) = {} & (2 \cdot \mathbf{x}_{s,k} + \dot{\mathbf{x}}_{s,k} - 2 \cdot \mathbf{x}_{s,k+1} + \dot{\mathbf{x}}_{s,k+1}) \cdot \bar{\tau}_k^3 \\ & + (-3 \cdot \mathbf{x}_{s,k} - 2 \cdot \dot{\mathbf{x}}_{s,k} + 3 \cdot \mathbf{x}_{s,k+1} - \dot{\mathbf{x}}_{s,k+1}) \cdot \bar{\tau}_k^2 + \dot{\mathbf{x}}_{s,k} \cdot \bar{\tau}_k + \mathbf{x}_{s,k} \end{aligned} \tag{14}$$

Similar to all other discretization schemes, the calculation of analytic gradient information can improve the convergence stability and speed of the discretized optimal control problem. As the slow dynamics are solved using a regular collocation scheme, the calculation method of the gradients does not change. The multiple shooting scheme for the fast states is slightly changed as the interpolated slow states have to be added to the equations. Consequently, the respective sensitivity equations need to be adapted accordingly. [5]

The method presented here can be used with a combination of explicit and implicit numerical integration schemes. Therein, explicit algorithms as well as implicit ones can easily be incorporated in the collocation, while for the shooting part the use of implicit methods comes at the cost of a very high computational burden. In this approach, path constraints can easily be calculated from the state histories on the coarse grid, as they are part of the optimization parameter vector. Anyway, it may happen that the states in between two such nodes violate a path constraint which may not be detected. [5]

5 EXAMPLE AND RESULTS

An air race trajectory optimization problem using a high fidelity dynamic model of an aerobatic aircraft is presented and solved here. The race track to be passed by the aircraft in minimum time is built up by seven race gates that are represented by the red crosses in figure 2. As the aircraft has to pass the gates several times, an optimal control problem with 17 point constraints and 16 intermediate phases results.

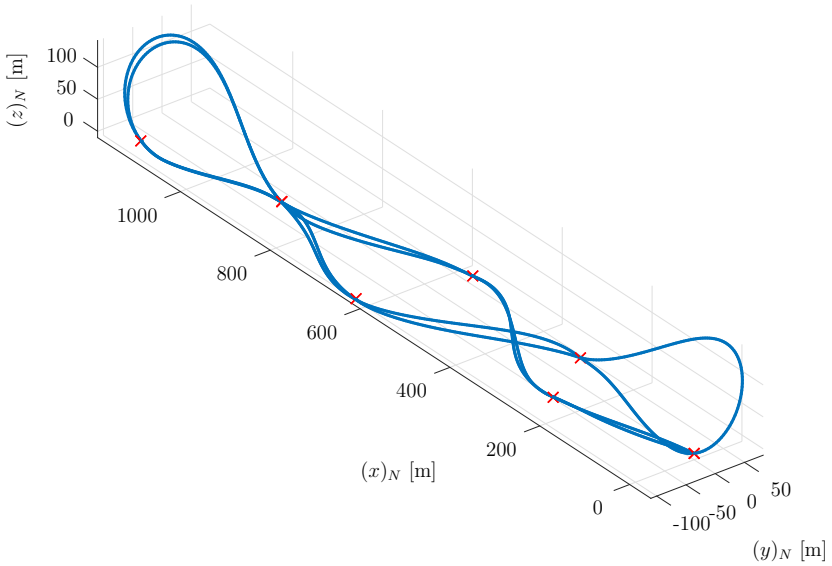

Figure 2: Time optimal spatial trajectory for the combined simulation model. [5]

5.1 Cost function

The cost function to be minimized in the air race example is the overall race time:

$$J = t_f - t_0 \tag{15}$$

which may be formulated as the LAGRANGE part of the cost function (1) as:

$$L(\mathbf{x}(t), \mathbf{u}(t), t) = 1 \tag{16}$$

5.2 Path constraints

Besides the dynamic constraints from the aircraft simulation model presented in section 3, algebraic path constraints have to hold along the trajectory. Table 2 lists these path constraints together with their respective limits. [5]

5.3 Race Gate Constraints

The constraints that apply whenever the aircraft is passing through one of the race gates are listed in table 3. [5]

Table 2: Path constraints applied in the air race example. [5]

Description	Symbol	Unit	Lower Bound	Upper Bound
Altitude limit due to ground clearance (zero-level at center of race gates)	$(h)_N$	m	-10	–
Load factor limits due to regulations	$(n_z)_B$	–	-2	10
Speed limits due to regulations	V_K	m/s	25	102.9
Aerodynamic angle of attack limits	α_A	deg	-10	20
Aerodynamic angle of sideslip limits	β_A	deg	-10	10
Roll rate limits	p_K	deg $/s$	-420	420

Table 3: Constraints applied at the race gates in the air race example. [5]

Description	Symbol	Unit	Tolerance
x-Position at the race gate	$(x)_N$	m	± 0
y-Position at the race gate	$(y)_N$	m	± 0
z-Position at the race gate	$(z)_N$	m	± 0
Orientation at the race gate	χ_K	deg	± 0
Bank angle at the race gate	μ_K	deg	± 10

5.4 Problem Setup

Here, the rotational states are considered to be fast, while the translational ones represent the slow dynamics of the aircraft. Consequently, the fast states in the model from above are

$$\mathbf{x}_f = [\mu_K, \alpha_K, \beta_K, p, q, r]^\intercal \tag{17}$$

while the remaining states are considered to be slow:

$$\mathbf{x}_s = [(x)_N, (y)_N, (z)_N, \chi_K, \gamma_K, V_K, \delta_T]^\intercal \tag{18}$$

In this example, the slow states were discretized on an equidistant grid in normalized time, containing 101 points per phase. For the fast states a subdivision of five grid points was used, resulting in a total of 501 grid points per phase. Overall, in the problem built up from 16 phases, for the slow states 1616 collocation nodes and for the fast states 8016 EULER forward multiple shooting grid points result. For the approximation of the slow states on the grid of the fast states, the linear interpolation from equation (11) has been implemented. The controls are also discretized on the coarse collocation grid and are linearly interpolated to the fine multiple shooting integration grid.

Due to the different lengths in real time of the phases, the distance between two nodes varies for the slow states from $0.024s$ to $0.065s$ and for the fast states from $0.0048s$ to $0.0129s$. *IPOPT* [12] was used to solve the problem to a feasibility tolerance of $tol_C = 10^{-6}$ and an optimality tolerance of $tol_J = 5 \cdot 10^{-6}$. [5]

5.5 Implementation in FALCON.m

The optimal control tool *FALCON.m* (*FSD Optimal Control Tool*) was used here. It has been developed at the Institute of Flight System Dynamics at the Technical University of Munich in order to avoid the need to re-implement discretization code for optimal control problems. After the user implemented all vectors and functions defining the problem, *FALCON.m* automatically calculates a discretized representation of it and uses *IPOPT* [12] to solve it. *FALCON.m* calculates all gradients analytically and determines the sparsity structure of the problem before handing this information to the solver for improved performance. [5]

5.6 Results

The problem was solved on a personal computer equipped with an Intel Core i7-950 CPU with $3.07GHz$ and $20GB$ of RAM and converged in approximately $2144s$ and 1355 iterations. The trajectory that results from the optimization can be seen in figure 2. The minimal race time calculated in the optimization is $60.3512s$. The state histories along the trajectories are depicted in figure 3.

For the sake of comparison, other optimizations have been performed using the simplified point mass simulation model as well as the rigid body simulation model – both in combination with a pure collocation algorithm. Table 4 gives a numeric overview of the different results. It can be seen that the number of optimization variables decreased to 27488 while the number of constraints dropped to 22627 in the discretized problem using the combined discretization scheme as less grid points are required. Even though the number of 764082 structural non-zero elements in the gradient is also comparably low, the sparsity ratio of 99.8772% is not as high as it is with pure collocation – as expected due to the higher interdependence in the shooting scheme and the smaller overall problem size. Additionally, the solution time also decreased, however, not in the same relation as the problem dimensions did, for the same reason as before.

Table 4: Overview of the results using the different simulation models and discretization methods. [5]

	Optim. param.	**Constr.**	**Iter.**	**Solution time**	**Cost func.**
Point mass simulation model (fine grid)	80176	72184	1927	$1730s$	$60.23s$
Combined collocation and shooting	27488	22627	1355	$2144s$	$60.35s$
Rigid body simulation model (fine grid)	136288	112227	883	$2290s$	$60.34s$

The yellow line in figure 4 shows the difference between the mixed collocation and shooting approach and the pure collocation of the rigid body simulation model on the fine grid. The observed difference in the trajectory is within the range of some centimeters. Here, the reason is twofold:

- On the one hand, the discretization of the controls is coarser with the combined solution method, reducing the theoretical "possibilities of the pilot", and

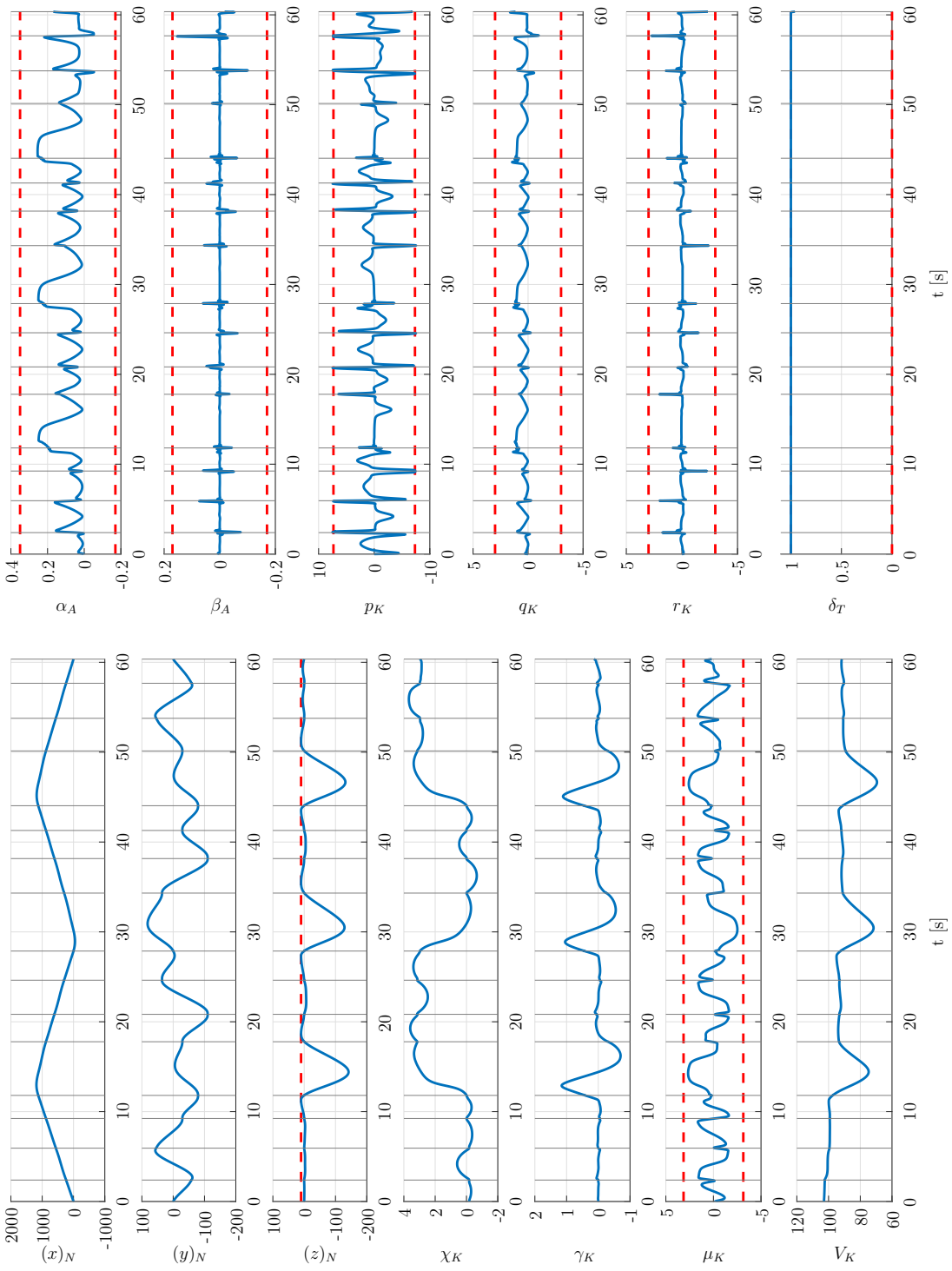

Figure 3: Optimal state histories for the combined simulation model. [5]

- on the other hand, in the two approaches different integration schemes are used for the fast dynamics which influence the numerical solution of the differential equation.

Anyway, the differences in the results achieved are expected to be much lower than any possible errors due to model inaccuracies. [5]

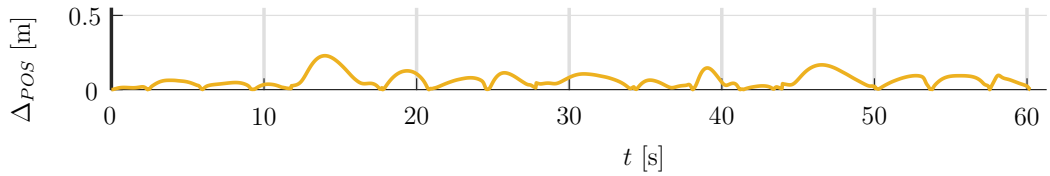

Figure 4: Position difference with respect to the solution for the rigid body simulation model. [5]

6 CONCLUSIONS

In the paper at hand, a novel discretization method for the solution of high fidelity optimal control problems incorporating different timescales is presented. The method splits the dynamic equations of the underlying system into a fast and a slow part. The slow dynamics are solved using a direct collocation scheme based on a relatively coarse grid. Therein, explicit or implicit integration methods may be used, as all state and control values are available in the optimization parameter vector of the discretized problem. The fast states are handled using a multiple shooting approach, with the grid points of the multiple shooting defects in time being equal to the collocation nodes of the slow dynamics. As the multiple shooting method allows for the use of an arbitrary number of integration steps between two defect nodes, the integration grid of the fast states may be chosen finer without increasing the problem dimensions. The multiple shooting approach results in a more dense gradient matrix of the overall problem due to its more nonlinear coupling of the states and controls. The calculation of the slow states between two collocation nodes – that are required to evaluate the fast dynamics – can be done using an arbitrary interpolation method, where in this paper linear and cubic interpolation is suggested. The method is also published as a part of the dissertation thesis of the first author [5].

The application of the method is demonstrated using an exemplary air race trajectory optimization problem incorporating a high fidelity rigid body simulation model with nonlinear aerodynamics. The results show that the problem size can be significantly reduced compared to a full discretization of the high fidelity problem. Similarly, the solution time decreases.

In the examples, a control discretization based on the coarse grid of the method is used, which is not ideal for all the controls as the control surface deflections ξ_{CMD}, η_{CMD} and ζ_{CMD} mainly influence the fast rotational dynamics and consequently also would require faster control inputs. Therefore, a future extension of the algorithm may be the use of a control grid which is similar to the *Multiple Shooting Integration Grid*. Thus, the problem

size will be increased by the additional control parameters without increasing the number of constraints. However, the calculation of the gradient matrix will change in this case, requiring further implementation efforts.

Besides, the combined discretization scheme presented here may be extended in future research by e. g. estimating the integration errors of the simulation in the *Multiple Shooting* segments and using this information for a grid refinement scheme. This way, the number of collocation and shooting nodes may be reduced, further decreasing problem size while increasing solution accuracy and the robustness of the solution process. [5]

REFERENCES

[1] S. Osher and R. Sanders, "Numerical approximations to nonlinear conservation laws with locally varying time and space grids," *Mathematics of Computation*, vol. 41, p. 321, 1983.

[2] M. Günther, A. Kværnø, and P. Rentrop, "Multirate partitioned runge-kutta methods," *BIT Numerical Mathematics*, vol. 41, no. 3, pp. 504–514, 2001.

[3] P. N. Desai and B. A. Conway, "Two-timescale discretization scheme for collocation," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1316–1322, 2008.

[4] C. L. Bottasso and G. Maisano, "Efficient rotorcraft trajectory optimization using comprehensive vehicle models by improved shooting methods," in *35th European Rotorcraft Forum Proceedings* (I. Lopez and P. Brandt, eds.), (Red Hook, NY), Curran Associates, 2009.

[5] M. Bittner, *Utilization of Problem and Dynamic Characteristics for Solving Large Scale Optimal Control Problems*. Dissertation, Technische Universität München, München, Submitted 01/18/2017, to be published.

[6] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Advances in Design and Control, Philadelphia: SIAM, Society for Industrial and Applied Mathematics, second edition ed., 2009.

[7] F. Fisch, *Development of a Framework for the Solution of High-Fidelity Trajectory Optimization Problems and Bilevel Optimal Control Problems*. München: Verlag Dr. Hut, 1. aufl. ed., 2011.

[8] L. Höcht, *Advances in Stability Analysis for Model Reference Adaptive Control Systems and Application to Unmanned Aerial Systems*. Dissertation, Technische Universität München, München, 2014.

[9] J. Lenz, *Optimisation of Periodic Flight Trajectories*. PhD thesis, Technische Universität München, München, 2015.

[10] R. Brockhaus, W. Alles, and R. Luckner, *Flugregelung*. Heidelberg, Dordrecht, London, New York: Springer, 3., neu bearb. aufl. ed., 2011.

[11] H. Weirather, *Flugrennsimulation gegen ein zeitoptimiertes Flugzeug*. Bachelor thesis, Technische Universität München, Garching, 2014.

[12] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.